

TEAP, LS 2020/2021
midterm, 14. 4. 2021

1. Na vstupe máme pole a , ktoré obsahuje n rôznych celých čísel, utriedených vzostupne. Navrhните algoritmus, ktorý v čase $O(\log n)$ zistí, či existuje i , pre ktoré platí $a[i] == i$. Detailne zdôvodnite jeho správnosť.
2. Napísal som takúto procedúru (predpokladám globálne pole a dĺžky n , ktoré obsahuje nuly a jednotky):

```
int huh() {
    int r = 0;
    for (int i = 0; i < n;)
        if (a[i] > 0) {
            int j = i;
            while (j < n - 1 && a[j + 1] > 0) j++;
            r += j - i + 1;
            i = j + 1;
        } else
            i++;
    return r;
}
```

Popíšte slovami, čo procedúra `huh` ráta a odvoďte jej zložitosť. Ako sa zmení zložitosť, ak príkaz `i = j + 1` nahradíme príkazom `i ++`?

3. Máme n vecí s neznámymi hmotnosťami. Naším cieľom je pomocou porovnávnia zistiť, ktorá vec koľko váži. Navrhните algoritmus, ktorý spracováva postupnosť m udalostí, pričom každá udalosť je buď tvaru `!i j x` alebo `?i j`. Udalosť `!i j x` znamená, že sme porovnali i tu a j tu vec a zistili sme, že j ta vec je o x gramov ťažšia. Pre každú udalosť `?i j` treba vypísať, o koľko gramov je j ta vec ťažšia ako i ta vec (alebo odpovedať, že sa to z doterajších meraní nedá zistiť). Ako vždy, dokážte korektnosť a odvoďte zložitosť. Snažte sa dosiahnuť čo najlepšiu zložitosť.

Napríklad pre vstup:

```
n=4 m=7
! 1 2 100
? 2 3
! 2 3 100
? 2 3
? 1 3
! 4 3 150
? 4 1
```

je výstup:

```
UNKNOWN
100
200
-50
```

4. Máme takúto procedúru:

```
int hroch(int k, int n) {
    if (k == 1) return (n + 1) / 2;
    if (n == 1) return 0;
    return 1 + hroch(k, hroch(k - 1, n));
}
```

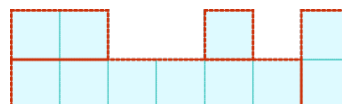
Nájdite najmenšie n také, že $\text{hroch}(3, n) \geq 7$.

-
5. (**bonusová úloha**) Máme n stĺpcov štvorcov, v i tom stĺpci je na seba naukladaných $a[i]$ štvorcov. Chceme ich pokryť obdĺžnikmi šírky alebo výšky 1 a pritom použiť čo najmenej obdĺžnikov. Navrhните algoritmus s čo najlepšou zložitosťou, dokážte jeho správnosť a odvoďte zložitosť.

Napríklad pre vstup:

```
2 2 1 1 2 1 2
```

je výstup 4, lebo nám stačí použiť 4 obdĺžniky napr. takto:



a tromi obdĺžnikmi to pokryť nevieme.